

MO417 - Complexidade de Algoritmos

Ata do exercício 22.4-2

Aula do dia 21/05/2009

Redatora: Ana Carolina Correia Rézio.

Enunciado:

Forneça um algoritmo de tempo linear que tome como entrada um grafo acíclico orientado $G=(V,E)$ e dois vértices s e t , e retorne o número de caminhos de s para t em G . Por exemplo, no grafo acíclico orientado da Figura 22.8, existem exatamente quatro caminhos do vértice p para o vértice v : pov , $poryv$, $posryv$ e $psryv$. (Seu algoritmo só precisa contar os caminhos, não listá-los.)

Resolução:

O seguinte algoritmo foi apresentado como solução:

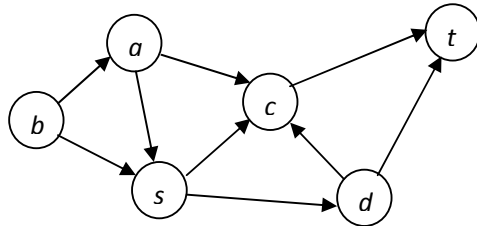
1. $T[] \leftarrow \text{OrdenacaoTopologica}(G)$ //Vetor T recebe os vértices ordenados topologicamente.
2. para cada $v \in V(G)$ faça
3. $v.\text{cont} = 0$
4. $\text{pos}_s \leftarrow \text{Busca}(T, s)$ //retorna a posição do vértice s no vetor T .
5. $\text{pos}_t \leftarrow \text{Busca}(T, t)$ //retorna a posição do vértice t no vetor T .
6. $s.\text{cont} = 1$
7. para $i \leftarrow \text{pos}_s$ até $\text{pos}_t - 1$ faça
8. $u \leftarrow T[i]$
9. para cada $v \in \text{Adj}[u]$ faça
10. $v.\text{cont} = v.\text{cont} + u.\text{cont}$
11. return $t.\text{cont}$

Análise do algoritmo

A linha 1 (os vértices são ordenados topologicamente e inseridos no vetor) possui custo $O(V+E)$. As linhas 2 e 3 possuem juntas o custo $\Theta(V)$. A linha 4 possui custo $O(V)$, o mesmo custo apresentado na linha 5. A linha 6 é $\Theta(1)$ e os passos presentes nas linhas 7 a 10 possuem, no pior caso (quando todos os vértices são visitados), o custo $O(V + E)$. Portanto a complexidade total do algoritmo é $O(V + E)$ - **Algoritmo Linear**.

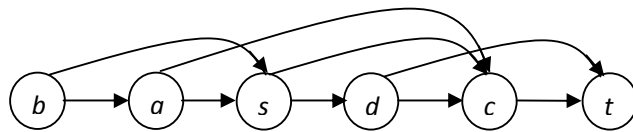
Ilustração

Considere o grafo G acíclico orientado:



A solução apresentada pelo algoritmo proposto para o grafo ilustrado segue:

a) Ordenação topológica



b) Associar valor 0 para todos os vértices e valor 1 para o vértice s.

b	a	s	d	c	t
0	0	1	0	0	0

c) Contar caminhos (linhas 7 a 10)

Iteração	b	a	s	d	c	t
1	0	0	1	1	1	0
2	0	0	1	1	2	1
3	0	0	1	1	2	3

Logo, existem 3 caminhos que saem do vértice s e chegam ao vértice t. Note que a iteração inicia sempre no vértice de origem s e finaliza no vértice anterior ao vértice de destino t, seguindo o vetor ordenado topologicamente. Caso a posição do vértice de origem no vetor seja maior do que o vértice de destino, o algoritmo retorna o número de caminhos igual a zero (devido a inicialização presente nas linhas 2 e 3).

Referências

[1] Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C.; Algoritmos: Teoria e Prática. Tradução da 2ª edição americana, 2002